REMARKS

Claims 1-18 are pending in the present application. In the above amendments, claims 15-18 have been amended merely to correct typographical errors with regard to claim dependency—no new issues have been raised, and no new matter has been added. Further, the specification has been amended, as is discussed below. Therefore, after entry of the above amendments, claims 1-18 will be pending in this application. Applicants believe that the present application is now in condition for allowance, which prompt and favorable action is respectfully requested.

Objection to the Specification

The Examiner has objected to the specification and requested updating of the cited reference to related applications. The Applicants have amended the specification in two places, as noted above, to include the respective application serial number and delete the Attorney Docket Number. As such, this objection is now moot. Therefore, the Applicants respectfully request the Examiner to withdraw the objection to the specification.

Rejection under 35 USC § 112, second paragraph

The Examiner has rejected claims 1, 6, 10 and 14 under 35 USC § 112, second paragraph. In particular, the Examiner states: "Each of applicants' independent claims recite the lmitation (sic), [']wherein the privileged function is executed as part of the same execution thread as the application.' "[1] The Applicants note that these claims actually recite: "wherein the function is executed within the same task as the application." In any case, the Applicants respectfully traverse this rejection.

[1] Office Action mailed March 14, 2006, page 2, paragraph #4—the sentences are transposed, but this appears to be what the Examiner is stating.

The Applicants respectfully submit that this rejection is in error. In particular, the Office Action states that "[a]s best the examiner can determine, applicant's system switches from executing the user thread, to executing the privileged thread, and back to executing the user thread. This does not seem to be a situation where one of ordinary skill would recognize the privilege function being executed as part of the user thread."[2] Further, the Examiner cites US Patent No. 6,175,916 and a definition of "thread" therein to support this position. Fundamentally, the '916 patent defines a "thread" in a manner suitable for its own purposes. This is permitted, as the principle that each patentee is entitled to be their own lexicographer is well settled. However, the limitations accorded to a "thread" in the '916 patent cannot be automatically imputed to any subsequent patent application. To do so would be to preclude any inventions which would improve upon the described subject matter by making such subject matter more flexible by removing one or more of such limitations, as in the present application.

Additionally, it should be noted that under many processor architectures, the user mode is associated with a first stack while the privileged mode is associated with a second stack.[3] Thus, in many processor architectures, switching between user and privileged modes necessarily requires switching between stacks, and transitioning from user to privileged to user modes would require saving the user mode stack pointers to permit the transition back to user mode without a loss of information. For these reasons, the rejection under 35 USC § 112, second paragraph, should be withdrawn.

Further, it should be noted that the definition of a "thread" in the '916 patent is incomplete. For example, the passage immediately after the referenced passage, i.e. column 1, lines 46-58 of the '916 patent, describe the operation of a conventional scheduler, and in

[2] *Id.* at page 3, lines 7-11.

[3] *See., e.g.,* Specification at Fig. 2 (code and data memories divided into independent partitions for user and privileged modes).

particular, implies that threads require scheduling for execution. Further, column 2, lines 16-31 of the '916 patent state that in a conventional system, a privilege mode task would be executed by a "system function" which is separately scheduled.

Yet, one important aspect of the present invention is the ability for one thread to switch from user mode to privilege mode and then back to user mode, without requiring the creation of a privileged mode thread, thereby avoiding the overhead associated with task creation.[4] The application therefore maintains consistency between what is described in its specification and the subject matter sought to be claimed. Accordingly, this provides an additional basis supporting the withdrawal of the rejection under 35 USC § 112, second paragraph.

Thus, for the reasons stated above, the Applicants respectfully request the Examiner to withdraw the rejection of claims 1, 6, 10 and 14 under 35 USC § 112, second paragraph.

Rejection under 35 USC § 101

The Examiner has rejected claims 1-18 under 35 USC § 101 "because the claims although tangible but (sic) lacks usefulness unless it (sic) disclose such an element."[5] The Applicants respectfully traverse the Examiner's assertion of a lack of usefulness/utility.

The Applicants submit that the present invention has a well-established specific, substantial and credible utility, as is immediately understood by one skilled in the art. In particular, the invention allows privileged mode functions to be invoked from applications executing in either a privileged or non-privileged mode. The desirability and usefulness of such an invention is described in more detail in the specification.[6] While the Examiner's suggested language provides a further explanation of the invention, the existing claims already recite that

[4] *See, e.g.* Specification at paragraph 39.

[5] *Id.* at page 3, paragraph #5.

[6] *See, e.g.*, Specification at page 2, paragraph 5 and at page 6, paragraphs 24 and 25.

an application executing in either a privileged or non-privileged mode has access to the function executing in the privileged mode. As such, the Examiner's suggested language is already supported by the existing claims. Thus, since the present invention has a well-established utility and since the proposed language is already supported by the claims, the Applicants respectfully request that the Examiner withdraw the rejection of claims 1-18 under 35 USC § 101 for lacking usefulness.

Rejection under 35 USC § 102(e)

The Examiner has rejected claims 1-18 under 35 USC § 102(e) as being anticipated by US Patent Pub. No. 2003/140245 to Dahan et al. The Applicants respectfully traverse this rejection.

In particular, Dahan does not disclose or suggest a method, apparatus or computer readable medium comprising any application executing in a non-privileged mode which invokes a function using an interrupt, where the function executes in the privileged mode, and further executing the function within the same task as the application, as recited in independent claims 1, 6, 10 and 14.

The Examiner alleges that Dahan discloses that the privileged function is executed within the same task as the application, however, Dahan in fact provides no such disclosure. In the only discussion of "tasks" within the reference, Dahan discloses that "[m]ost processors are built with two levels of privilege: one for the OS, and another one for user tasks,"[7] thereby implying that each task is associated with a distinct privilege level. Further, Dahan states that the invention of Dahan provides "a secure mode (3rd level of privilege) built in a non-invasive way on a processor

[7] US Patent Pub. No. 2003/0140245, page 1, paragraph 6.

system."[8] Therefore, as the secure mode of Dahan provides a 3rd level of privilege, it follows from the teaching of Dahan that any task executing in the secure mode must, by definition, be different from a user task based on the different level of privilege. Thus, Dahan does not disclose or suggest the recited claim language, and in fact teaches away from executing a function in the privileged mode within the same task as an application executing within a non-privileged mode, as recited by the claims.

Additionally, Dahan discloses that the secure mode of operation, i.e. the 3rd level of privilege, can only be entered through an "activation sequence," which is a piece of code that is executed to access a secure service.[9] Significantly, the execution of the activation sequence means that execution has switched away from the initial task in the user application and to a different task.

Thus, Dahan discloses that when an application operating in a non-privileged mode calls a service in a privileged level of operation, a new task is created to execute the activation sequence. Further, Dahan teaches that the secure mode of operation provides a 3rd level of privilege, which by Dahan's own teaching is a different task than the user task executing the non-privileged application. Therefore, Dahan does not disclose or suggest, and in fact teaches away from, an application executing in a non-privileged mode which invokes a function using an interrupt, where the function executes in the privileged mode, and further executing the function within the same task as the application, as recited in independent claims 1, 6, 10 and 14.

For the same reasons as above, claims 2-5, 7-9, 11-13 and 15-18 are distinguishable over Dahan, as these claims depend from one of independent claims 1, 6, 10 and 14.

[8] *Id.* at page 1, paragraph 10, lines 2 and 3.

[9] *Id.* at page 5, paragraph 76, lines 3-6.

Therefore, based on the above remarks, the Applicants respectfully request the Examiner to withdraw the rejection of claims 1-18 under 35 USC § 102(e) as being anticipated by Dahan et al.

Rejection under 35 USC § 102(b)

The Examiner has rejected claims 1-2, 4, 6-7, 10-11, 14-15 and 17 under 35 USC § 102(b) as being anticipated by US Patent No. 6,175,916 to Ginsberg et al. The Applicants respectfully traverse this rejection.

The Office Action states that Ginsberg discloses a system in which a user mode thread executes a privileged function by transitioning to a privileged mode which executes in the same thread as the user mode thread, and specifically identifies three passages for support, namely: column 5, lines 1-23; column 6, line 65 – column 7, line 20; and column 8, line 50 to column 10, line 17.

1. Column 5, lines 1-23

Column 5, lines 1-23, describes processor 41 of the Ginsberg, which is disclosed to be a conventional off the shelf microprocessor, such as the MIPS R3000 processor. This passage further discloses that the microprocessor supports privileged and user modes, and the use of fault handlers. However, contrary to the assertion in the Office Action, this passage fails to disclose or suggest any user mode application which invokes a function using an interrupt, where the function executes in the privileged mode, and further executing the function within the same task as the application, as recited in independent claim 1. Independent claims 6, 10 and 14 recite similar subject matter.

2. Column 6, line 65 – column 7, line 20

Column 6, line 65 to column 7, line 20 describe how Ginsberg utilizes a jump instruction to an invalid address to trigger a memory fault, causing its operating system to execute a call handler, which will recognize a call to a system function if the invalid memory address is an odd address. However, contrary to the assertion in the Office Action, this passage fails to disclose or suggest any user mode application which invokes a function using an interrupt, where the function executes in the privileged mode, and further executing the function within the same task as the application, as recited in independent claim 1. Independent claims 6, 10 and 14 recite similar subject matter.

3. Column 8, line 50 to column 10, line 17

Column 8, line 50 to column 10, line 17 is part of a larger passage starting from column 7, line 21 to column 10, line 28, describing Fig. 9. The extended passage describes how a user process 100 may call a system process 102 in the following manner. First, the user process 100 creates a memory fault. Column 8, line 49-53. Responsive to the memory fault, a fault handler is invoked. Column 8, lines 52-60. The fault handler identifies the memory address which created the memory fault, and maps that address to a system call, and then calls the identified system call. Significantly, the invocation of the fault handler means that execution has switched away from the thread in the user process 100 which caused the memory fault and to a different thread. The passage further describes how a call processing function is utilized to process memory addresses prior to the fault handler calling the identified system call in the system process 102. Significantly, Ginsberg discloses that the fault handler, the call processing function, and the system function itself are executed within the same execution thread. Column 9, 65-67. Because the fault handler is invoked by the processor in response to the memory fault caused by the user thread in the user process 100, the user thread must be a different thread than that of the fault handler, call processing function, and system function.

In fact, not only does Ginsberg disclose using a separate thread (i.e., one comprising the fault handler, call processing function, and system function) to service the system call, the Ginsberg method even executes the system call in a non-privileged mode. For example, column 9, lines 54 – 57 states the "[f]ault handler 104 switches to the system address space of the specified system function, restores the previous execution mode (normally non-privileged mode) and calls the system function."

Thus, contrary to the assertion made in the Office Action, Ginsberg fails to disclose or suggest any user mode application which invokes a function using an interrupt, where the function executes in the privileged mode, and further executing the function within the same task as the application, as recited in independent claim 1. Independent claims 6, 10 and 14 recite similar subject matter.

Additionally, and for the same reasons, claims 2, 4, 7, 11, 15 and 17 are not disclosed or suggested by Ginsberg, as these claims dependent from respective ones of independent claims 1, 6, 10 and 14.

Therefore, based on the above remarks, the Applicants respectfully request the Examiner to withdraw the rejection of claims 1-2, 4, 6-7, 10-11, 14-15 and 17 under 35 USC § 102(b) as being anticipated by Ginsberg.

Rejection under 35 USC § 103(a)

The Examiner has rejected claims 5, 9, 13 and 18 under 35 USC § 103(a) as being obvious over Ginsberg in view of Applicant Admittance Prior Art (AAPA). The Applicants respectfully traverse this rejection.

The deficiencies of Ginsberg are documented in detail directly above, and similarly apply to claims 5, 9, 13 and 18 as they depend from respective ones of independent claims 1, 6, 10 and 14. Further, the addition of AAPA does not solve the deficiencies of Ginsberg, as AAPA does not disclose or suggest the recited subject matter. Thus, Ginsberg and AAPA, in any combination, do not disclose or suggest the subject matter recited by claims 5, 9, 13 and 18.

Therefore, the Applicants respectfully request the Examiner to withdraw the rejection of claims 5, 9, 13 and 18 under 35 USC § 103(a) as being obvious over Ginsberg in view of AAPA.

CONCLUSION

In light of the amendments contained herein, Applicants submit that the application is in condition for allowance, for which early action is requested.

Please charge any fees or overpayments that may be due with this response to Deposit Account No. 17-0026.

Respectfully submitted,

Dated June 14, 2006
:

By: /Robert J. O'Connell/

Robert J. O'Connell
Reg. No. 44,265
(858) 651-4361

QUALCOMM Incorporated
Attn: Patent Department
5775 Morehouse Drive
San Diego, California 92121-1714
Telephone: (858) 658-5787
Facsimile: (858) 658-2502